New
March 29, 2001
Tomohiko Ogishi et al.
0965-0343P
BSKB
703-205-8000

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２０００年 ３月２９日

出願番号
Application Number: 特願２０００－０９０６６１

出願人
Applicant (s): ケイディディ株式会社

２０００年１０月１３日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号 出証特２０００－３０８４０２３

【書類名】　特許願

【整理番号】　K00020901

【特記事項】　特許法第３０条第１項の規定の適用を受けようとする特許出願

【提出日】　平成12年 3月29日

【あて先】　特許庁長官殿

【国際特許分類】　H04L 12/28

H04Q 3/00

【発明者】

【住所又は居所】　埼玉県上福岡市大原二丁目１番１５号　株式会社ケイディディ研究所内

【氏名】　大岸　智彦

【発明者】

【住所又は居所】　埼玉県上福岡市大原二丁目１番１５号　株式会社ケイディディ研究所内

【氏名】　井戸上　彰

【発明者】

【住所又は居所】　埼玉県上福岡市大原二丁目１番１５号　株式会社ケイディディ研究所内

【氏名】　長谷川　亨

【発明者】

【住所又は居所】　埼玉県上福岡市大原二丁目１番１５号　株式会社ケイディディ研究所内

【氏名】　加藤　聰彦

【特許出願人】

【識別番号】　000001214

【氏名又は名称】　ケイディディ株式会社

【代理人】

【識別番号】　100078499

【弁理士】
【氏名又は名称】　光石　俊郎
【電話番号】　03-3583-7058
【選任した代理人】
【識別番号】　100074480
【弁理士】
【氏名又は名称】　光石　忠敬
【電話番号】　03-3583-7058
【選任した代理人】
【識別番号】　100102945
【弁理士】
【氏名又は名称】　田中　康幸
【電話番号】　03-3583-7058
【手数料の表示】
【予納台帳番号】　020318
【納付金額】　21,000円
【提出物件の目録】
【物件名】　明細書　1
【物件名】　図面　1
【物件名】　要約書　1
【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　トラヒック統計情報収集方法

【特許請求の範囲】

【請求項１】　インターネット回線上の一方向のトラヒックから、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属するＤＡＴＡセグメントを検出すること、及び、
発側転送セグメント数として、検出したＤＡＴＡセグメントの総数を計算し、あるいは、発側転送バイト数として、最初に検出したＤＡＴＡセグメントのシーケンス番号と、最後に検出したＤＡＴＡセグメントのシーケンス番号にそれのユーザデータ長を加算した値との差を計算し、あるいは、前記発側転送セグメント数及び前記発側転送バイト数の両方を検出すること
を特徴とするトラヒック統計情報収集方法。

【請求項２】　インターネット回線上の一方向のトラヒックから、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＤＡＴＡセグメントを検出すること、及び、
着側転送セグメント数として、検出したＤＡＴＡセグメントの総数を計算し、あるいは、着側転送バイト数として、最初に検出したＤＡＴＡセグメントのシーケンス番号と、最後に検出したＤＡＴＡセグメントのシーケンス番号にそれのユーザデータ長を加算した値との差を計算し、あるいは、前記着側転送セグメント数及び前記着側転送バイト数の両方を計算すること
を特徴とするトラヒック統計情報収集方法。

【請求項３】　インターネット回線上の一方向のトラヒックから、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属するＡＣＫセグメント及びＤＡＴＡセグメントを検出すること、及び、
発側データ受信量として、最初に検出したＡＣＫセグメント又はＤＡＴＡセグメントの応答確認番号と最後に検出したＡＣＫセグメント又はＤＡＴＡセグメントの応答確認番号との差を計算すること
を特徴とするトラヒック統計情報収集方法。

【請求項４】　インターネット回線上の一方向のトラヒックから、ＳＹＮ＋

ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＡＣＫセグメント及びＤＡＴＡセグメントを検出すること、及び、

着側データ受信量として、最初に検出したＡＣＫセグメント又はＤＡＴＡセグメントの応答確認番号と最後に検出したＡＣＫセグメント又はＤＡＴＡセグメントの応答確認番号との差を計算すること

を特徴とするトラヒック統計情報収集方法。

【請求項５】　インターネット回線上の一方向のトラヒックから、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属するＤＡＴＡセグメントを検出すること、

ＤＡＴＡセグメントの検出毎に、検出したＤＡＴＡセグメントのシーケンス番号から、次に送られてくるべきＤＡＴＡセグメントのシーケンス番号（以下、次シーケンス番号と呼ぶ）を求めること、

ＤＡＴＡセグメントの検出毎に、新たに検出したＤＡＴＡセグメントのシーケンス番号が前回のＤＡＴＡセグメントの検出で求めた次シーケンス番号より小さいか否か判定すること、及び、

前記判定が肯定である場合に、今までの発側再送セグメント数に１を加算して新たな発側再送セグメント数を求め、あるいは、今までの発側再送バイト数に、前記次シーケンス番号と前記新たに検出したＤＡＴＡセグメントのシーケンス番号との差と、前記新たに検出したＤＡＴＡセグメントのユーザデータ長とのうち、小さい方を加算して新たな発側再送バイト数を求め、あるいは、前記新たな発側再送セグメント数及び前記新たな発側再送バイト数の両方を求めること

を特徴とするトラヒック統計情報収集方法。

【請求項６】　インターネット回線上の一方向のトラヒックから、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＤＡＴＡセグメントを検出すること、

ＤＡＴＡセグメントの検出毎に、検出したＤＡＴＡセグメントのシーケンス番号から、次に送られてくるべきＤＡＴＡセグメントのシーケンス番号（以下、次シーケンス番号と呼ぶ）を求めること、

ＤＡＴＡセグメントの検出毎に、新たに検出したＤＡＴＡセグメントのシーケンス番号が前回のＤＡＴＡセグメントの検出で求めた次シーケンス番号より小さいか否か判定すること、及び、
前記判定が肯定である場合に、今までの着側再送セグメント数に１を加算して新たな着側再送セグメント数を求め、あるいは、今までの着側再送バイト数に、前記次シーケンス番号と前記新たに検出したＤＡＴＡセグメントのシーケンス番号との差と、前記新たに検出したＤＡＴＡセグメントのユーザデータ長とのうち、小さい方を加算して新たな着側再送バイト数を求め、あるいは、前記新たな着側再送セグメント数及び前記新たな着側再送バイト数の両方を求めること
を特徴とするトラヒック統計情報収集方法。

【請求項７】　インターネット回線上の一方向のトラヒックから、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属するＡＣＫセグメントを検出すること、
ＡＣＫセグメントの検出毎に、検出したＡＣＫセグメントの応答確認番号及びウインドウサイズから、次に送られてくるべきＡＣＫセグメントの応答確認番号（以下、次応答確認番号と呼ぶ）と、最大の応答確認番号を持つＡＣＫセグメントのウインドウサイズ（以下、現ウインドウサイズと呼ぶ）を求めること、
ＡＣＫセグメントの検出毎に、新たに検出したＡＣＫセグメントの応答確認番号及びウインドウサイズの両方が前回のＡＣＫセグメントの検出で求めた次応答確認番号及び現ウインドウサイズに等しいか否かを判定すること、及び、
応答確認番号及びウインドウサイズの組が同じ値の２つ以上のＡＣＫセグメントについて前記判定が肯定である毎に、今までの発側ＤＡＴＡセグメント紛失数に１を加算して新たな発側ＤＡＴＡセグメント紛失数を求めること
を特徴とするトラヒック統計情報収集方法。

【請求項８】　インターネット回線上の一方向のトラヒックから、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＡＣＫセグメントを検出すること、
ＡＣＫセグメントの検出毎に、検出したＡＣＫセグメントの応答確認番号及びウインドウサイズから、次に送られてくるべきＡＣＫセグメントの応答確認番号（

以下、次応答確認番号と呼ぶ）と、最大の応答確認番号を持つＡＣＫセグメントのウインドウサイズ（以下、現ウインドウサイズと呼ぶ）を求めること、
ＡＣＫセグメントの検出毎に、新たに検出したＡＣＫセグメントの応答確認番号及びウインドウサイズの両方が前回のＡＣＫセグメントの検出で求めた次応答確認番号及び前記現ウインドウサイズに等しいか否かを判定すること、及び、
応答確認番号及びウインドウサイズの組が同じ値の２つ以上のＡＣＫセグメントについて前記判定が肯定である毎に、今までの着側ＤＡＴＡセグメント紛失数に１を加算して新たな着側ＤＡＴＡセグメント紛失数を求めること
を特徴とするトラヒック統計情報収集方法。

【請求項９】　インターネット回線上の一方向のトラヒックから、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属する連続複数のＤＡＴＡセグメント及び同ＤＡＴＡセグメントに続く連続複数のＡＣＫセグメントを検出すること、及び、
発側ＨＴＴＰ応答時間として、最後のＤＡＴＡセグメントの検出からの最初のＡＣＫセグメント検出までの時刻差を計算し、あるいは、発側ＨＴＴＰスループットとして、最初に検出したＡＣＫセグメントと最後に検出したＡＣＫセグメントとの応答確認番号の差と、最初のＡＣＫセグメントの検出から最後のＡＣＫセグメントの検出までの時刻差の比を計算し、あるいは、前記発側ＨＴＴＰ応答時間及び前記発側ＨＴＴＰスループットの両方を計算すること
を特徴とするトラヒック統計情報収集方法。

【請求項１０】　インターネット回線上の一方向のトラヒックから、ＳＹＮ＋ＡＣＫセグメントを検出し，更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属する連続複数のＡＣＫセグメント及び同ＡＣＫセグメントに続く連続複数のＤＡＴＡセグメントを検出すること、
着側ＨＴＴＰ応答時間として、最後のＡＣＫセグメントの検出から最初のＤＡＴＡセグメントの検出までの時刻差を計算し、あるいは、着側ＨＴＴＰスループットとして、最初に検出したＤＡＴＡセグメントのシーケンス番号と最後に検出したＤＡＴＡセグメントのシーケンス番号にそれのユーザデータ長を加算した値との差と、最初のＤＡＴＡセグメントの検出から最後のＤＡＴＡセグメントの検出

までの時刻差との比を計算し、あるいは、前記着側ＨＴＴＰ応答時間及び前記着側ＨＴＴＰスループットの両方を計算すること
を特徴とするトラヒック統計情報収集方法。

【請求項１１】　インターネット回線上の一方向のトラヒックから、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属する連続複数のＡＣＫセグメントを検出すること、及び、
発側ＦＴＰスループットとして、最初に検出したＡＣＫセグメントと最後に検出したＡＣＫセグメントとの応答確認番号の差と、最初のＡＣＫセグメントの検出から最後のＡＣＫセグメントの検出までの時刻差との比を計算すること
を特徴とするトラヒック統計情報収集方法。

【請求項１２】　インターネット回線上の一方向のトラヒックから、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属する連続複数のＤＡＴＡセグメントを検出すること、及び、
着側ＦＴＰスループットとして、最初に検出したＤＡＴＡセグメントのシーケンス番号と最後に検出したＤＡＴＡセグメントのシーケンス番号にそれのユーザデータ長を加算した値との差と、最初のＤＡＴＡセグメントの検出から最後のＤＡＴＡセグメントの検出までの時刻差との比を計算すること
を特徴とするトラヒック統計情報収集方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明はインターネット回線上を流れるＴＣＰ／ＩＰのトラヒックをキャプチャ（取得）し、それを基にトラヒックの統計情報を収集する技術に関し、特に、片方向のトラヒックのみ取得できる場合において、ＴＣＰレベルの情報を管理することにより、トラヒックが取得できなかった方向のトラヒック統計情報の収集を可能にするものである。

【０００２】

【従来の技術】

インターネットの普及に伴い、データ転送量等、トラヒックの統計情報を測定

するパフォーマンス解析が重要となっている。これに応え、特開平１１－２５２１１１号公報に開示されたように、本発明者らは、ＴＣＰレベルのトラヒック統計情報（再送データ量やコネクション確立の待ち時間など）を測定することによりインターネットの状況を判別できると考え、バックボーン回線等をモニタしＩＰレベルのトラヒック統計情報に加えて、ＴＣＰレベルも併せて収集することが可能なトラヒック監視装置を既に開発した。このトラヒック監視装置は、ＴＣＰ通信の双方向のセグメントが同一回線でモニタ可能であることを前提する。

【０００３】

しかしながら、ＩＰルーチングは方向別に制御されるため、必ずしも双方向のトラヒックが同一回線を経由するとは限らない。つまり、現実のネットワーク構成においては、行き帰りのルートが違う場合が考えられるから、必ずしも双方向のトラヒックを取得できるとは限らない。

【０００４】

その他、ルータにおいて単位時間毎のセグメント数やバイト数などのＩＰレベルのトラヒック統計情報を収集する技術（ＭＲＴＧと呼ばれる）が知られている。また、ＩＰレベルのトラヒック統計情報に加え、アプリケーション別のセグメント数やバイト数などのＴＣＰレベルのトラヒック統計情報を収集する技術（ＳＮＩＦＦＥＲと呼ばれる）も知られている。しかし、いずれもＴＣＰ通信の双方向のセグメントが同一回線でモニタ可能であることを前提とする。

【０００５】

【発明が解決しようとする課題】

本発明の目的は、片方向のトラヒックしか取得できない場合においても、そのトラヒックを用いてＴＣＰレベルの情報を管理することにより、双方向、特に、トラヒックが取得できなかった方向のトラヒック統計情報を収集することを可能にすることである。

【０００６】

【課題を解決するための手段】

請求項１に係る発明は、インターネット回線上の一方向のトラヒックからセグメントを取得し、発側データ転送量を収集するトラヒック統計情報収集方法であ

り、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属するＤＡＴＡセグメントを検出し、発側転送セグメント数として、検出したＤＡＴＡセグメントの総数を計算し、あるいは、発側転送バイト数として、最初に検出したＤＡＴＡセグメントのシーケンス番号と、最後に検出したＤＡＴＡセグメントのシーケンス番号にそれのユーザデータ長を加算した値との差を計算し、あるいは、前記発側転送セグメント数及び前記発側転送バイト数の両方を検出する。

請求項２に係る発明は、着側データ転送量を収集するトラヒック統計情報収集方法であり、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＤＡＴＡセグメントを検出し、着側転送セグメント数として、検出したＤＡＴＡセグメントの総数を計算し、あるいは、着側転送バイト数として、最初に検出したＤＡＴＡセグメントのシーケンス番号と、最後に検出したＤＡＴＡセグメントのシーケンス番号にそれのユーザデータ長を加算した値との差を計算し、あるいは、前記着側転送セグメント数及び前記着側転送バイト数の両方を計算する。

【０００７】

請求項３に係る発明は、インターネット回線上の一方向のトラヒックからセグメントを取得し、発側データ受信量を収集するトラヒック統計情報収集方法であり、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属するＡＣＫセグメント及びＤＡＴＡセグメントを検出し、発側データ受信量として、最初に検出したＡＣＫセグメント又はＤＡＴＡセグメントの応答確認番号と最後に検出したＡＣＫセグメント又はＤＡＴＡセグメントの応答確認番号との差を計算する。

請求項４に係る発明は、着側データ受信量を収集するトラヒック統計情報収集方法であり、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＡＣＫセグメント及びＤＡＴＡセグメントを検出し、着側データ受信量として、最初に検出したＡＣＫセグメント又はＤＡＴＡセグメントの応答確認番号と最後に検出したＡＣＫセグメント又はＤＡＴＡセグメントの応答確認番号との差を計算する。

【０００８】

請求項５に係る発明は、インターネット回線上の一方向のトラヒックからセグメントを取得し、発側再送量を収集するトラヒック統計情報収集方法であり、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属するＤＡＴＡセグメントを検出し、ＤＡＴＡセグメントの検出毎に、検出したＤＡＴＡセグメントのシーケンス番号から、次に送られてくるべきＤＡＴＡセグメントのシーケンス番号（以下、次シーケンス番号と呼ぶ）を求め、ＤＡＴＡセグメントの検出毎に、新たに検出したＤＡＴＡセグメントのシーケンス番号が前回のＤＡＴＡセグメントの検出で求めた次シーケンス番号より小さいか否か判定し、前記判定が肯定である場合に、今までの発側再送セグメント数に１を加算して新たな発側再送セグメント数を求め、あるいは、今までの発側再送バイト数に、前記次シーケンス番号と前記新たに検出したＤＡＴＡセグメントのシーケンス番号との差と、前記新たに検出したＤＡＴＡセグメントのユーザデータ長とのうち、小さい方を加算して新たな発側再送バイト数を求め、あるいは、前記新たな発側再送セグメント数及び前記新たな発側再送バイト数の両方を求める。

請求項６に係る発明は、着側再送量を収集するトラヒック統計情報収集方法であり、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＤＡＴＡセグメントを検出し、ＤＡＴＡセグメントの検出毎に、検出したＤＡＴＡセグメントのシーケンス番号から、次に送られてくるべきＤＡＴＡセグメントのシーケンス番号（以下、次シーケンス番号と呼ぶ）を求め、ＤＡＴＡセグメントの検出毎に、新たに検出したＤＡＴＡセグメントのシーケンス番号が前回のＤＡＴＡセグメントの検出で求めた次シーケンス番号より小さいか否か判定し、前記判定が肯定である場合に、今までの着側再送セグメント数に１を加算して新たな着側再送セグメント数を求め、あるいは、今までの着側再送バイト数に、前記次シーケンス番号と前記新たに検出したＤＡＴＡセグメントのシーケンス番号との差と、前記新たに検出したＤＡＴＡセグメントのユーザデータ長とのうち、小さい方を加算して新たな着側再送バイト数を求め、あるいは、前記新たな着側再送セグメント数及び前記新たな着側再送バイト数の両方を求める。

【０００９】

請求項７に係る発明は、インターネット回線上の一方向のトラヒックからセグメントを取得し、発側ＤＡＴＡセグメント紛失数を収集するトラヒック統計情報収集方法であり、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属するＡＣＫセグメントを検出し、ＡＣＫセグメントの検出毎に、検出したＡＣＫセグメントの応答確認番号及びウインドウサイズから、次に送られてくるべきＡＣＫセグメントの応答確認番号（以下、次応答確認番号と呼ぶ）と、最大の応答確認番号を持つＡＣＫセグメントのウインドウサイズ（以下、現ウインドウサイズと呼ぶ）を求め、ＡＣＫセグメントの検出毎に、新たに検出したＡＣＫセグメントの応答確認番号及びウインドウサイズの両方が前回のＡＣＫセグメントの検出で求めた次応答確認番号及び現ウインドウサイズに等しいか否かを判定し、応答確認番号及びウインドウサイズの組が同じ値の２つ以上のＡＣＫセグメントについて前記判定が肯定である毎に、今までの発側ＤＡＴＡセグメント紛失数に１を加算して新たな発側ＤＡＴＡセグメント紛失数を求める。

請求項８に係る発明は、着側ＤＡＴＡセグメント紛失数を収集するトラヒック統計情報収集方法であり、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＡＣＫセグメントを検出し、ＡＣＫセグメントの検出毎に、検出したＡＣＫセグメントの応答確認番号及びウインドウサイズから、次に送られてくるべきＡＣＫセグメントの応答確認番号（以下、次応答確認番号と呼ぶ）と、最大の応答確認番号を持つＡＣＫセグメントのウインドウサイズ（以下、現ウインドウサイズと呼ぶ）を求め、ＡＣＫセグメントの検出毎に、新たに検出したＡＣＫセグメントの応答確認番号及びウインドウサイズの両方が前回のＡＣＫセグメントの検出で求めた次応答確認番号及び前記現ウインドウサイズに等しいか否かを判定し、応答確認番号及びウインドウサイズの組が同じ値の２つ以上のＡＣＫセグメントについて前記判定が肯定である毎に、今までの着側ＤＡＴＡセグメント紛失数に１を加算して新たな着側ＤＡＴＡセグメント紛失数を求める。

【００１０】

請求項９に係る発明は、インターネット回線上の一方向のトラヒックからセグメントを取得し、発側のＨＴＴＰレベルの情報を収集するトラヒック統計情報収集方法であり、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属する連続複数のＤＡＴＡセグメント及び同ＤＡＴＡセグメントに続く連続複数のＡＣＫセグメントを検出し、発側ＨＴＴＰ応答時間として、最後のＤＡＴＡセグメントの検出から最初のＡＣＫセグメントの検出までの時刻差を計算し、あるいは、発側ＨＴＴＰスループットとして、最初に検出したＡＣＫセグメントと最後に検出したＡＣＫセグメントとの応答確認番号の差と、最初のＡＣＫセグメントの検出から最後のＡＣＫセグメントの検出までの時刻差の比を計算し、あるいは、前記発側ＨＴＴＰ応答時間及び前記発側ＨＴＴＰスループットの両方を計算する。

請求項１０に係る発明は、着側のＨＴＴＰレベルの情報を収集するトラヒック統計情報収集方法であり、ＳＹＮ＋ＡＣＫセグメントを検出し，更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属する連続複数のＡＣＫセグメント及び同ＡＣＫセグメントに続く連続複数のＤＡＴＡセグメントを検出し、着側ＨＴＴＰ応答時間として、最後のＡＣＫセグメントの検出から最初のＤＡＴＡセグメントの検出までの時刻差を計算し、あるいは、着側ＨＴＴＰスループットとして、最初に検出したＤＡＴＡセグメントのシーケンス番号と最後に検出したＤＡＴＡセグメントのシーケンス番号にそれのユーザデータ長を加算した値との差と、最初のＤＡＴＡセグメントの検出から最後のＤＡＴＡセグメントの検出までの時刻差との比を計算し、あるいは、前記着側ＨＴＴＰ応答時間及び前記着側ＨＴＴＰスループットの両方を計算する。

上記いずれの場合も、前記ＡＣＫセグメントを連続して所定数以上、例えば１０以上検出した場合のみ、前記発側ＨＴＴＰスループットあるいは前記着側ＨＴＴＰスループットの計算を行うことにより、異常値を効果的に排除することができる。

【００１１】

請求項１１に係る発明は、インターネット回線上の一方向のトラヒックからセグメントを取得し、発側のＦＴＰレベルの情報を収集するトラヒック統計情報収

集方法であり、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属する連続複数のＡＣＫセグメントを検出し、発側ＦＴＰスループットとして、最初に検出したＡＣＫセグメントと最後に検出したＡＣＫセグメントとの応答確認番号の差と、最初のＡＣＫセグメントの検出から最後のＡＣＫセグメントの検出までの時刻差との比を計算する。この場合、前記ＡＣＫセグメントを連続して所定数以上、例えば１０以上検出した場合のみ、前記発側ＦＴＰスループットの計算を行うことにより、異常値を効果的に排除することができる。

請求項１２に係る発明は、着側のＦＴＰレベルの情報を収集するトラヒック統計情報収集方法であり、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属する連続複数のＤＡＴＡセグメントを検出し、着側ＦＴＰスループットとして、最初に検出したＤＡＴＡセグメントのシーケンス番号と最後に検出したＤＡＴＡセグメントのシーケンス番号にそれのユーザデータ長を加算した値との差と、最初のＤＡＴＡセグメントの検出から最後のＤＡＴＡセグメントの検出までの時刻差との比を計算する。この場合、前記ＤＡＴＡセグメントを連続して所定数以上、例えば１０以上検出した場合のみ、前記着側ＦＴＰスループットの計算を行うことにより、異常値を効果的に排除することができる。

【００１２】

【発明の実施の形態】

以下、図面を参照して、本発明の実施形態例を説明する。

【００１３】

図１に、本発明を実現する双方向トラヒック統計情報収集装置の構成例を示す。図１に示す装置１は、セグメント取得モジュール２と、解析モジュール３と、蓄積モジュール４と、状態遷移表の格納部５と、統計情報の格納部６を有している。この装置１が収集するトラヒック統計情報は、大別するとＴＣＰレベルの情報及びアプリケーションレベル（ＨＴＴＰレベル及びＦＴＰレベル）の情報であり、以下に詳細を示す。ここで、発側とはＳＹＮセグメントの送信側、着側とはＳＹＮ＋ＡＣＫセグメントの送信側を意味する。但し、ＦＴＰのデータ用ＴＣＰ

コネクションではサーバがＳＹＮセグメントを送信し、クライアントがＳＹＮ＋ＡＣＫセグメントを送信するので、ＦＴＰでの発側とはＳＹＮ＋ＡＣＫセグメントの送信側、着側とはＳＹＮセグメントの送信側を意味する。

(1) ＴＣＰレベルの情報には、下記のものがある。

発側データ転送量（セグメント数／バイト数）

着側データ転送量（セグメント数／バイト数）

発側データ受信量（バイト数）

着側データ受信量（バイト数）

発側再送量（セグメント数／バイト数）

着側再送量（セグメント数／バイト数）

発側ＤＡＴＡセグメント紛失数（セグメント数）

着側ＤＡＴＡセグメント紛失数（セグメント数）

(2) ＨＴＴＰレベルの情報には、下記のものがある。

発側応答時間

発側スループット

着側応答時間

着側スループット

(3) ＦＴＰレベルの情報には、下記のものがある。

発側スループット

着側スループット

【００１４】

セグメント取得モジュール２は、インターネット回線（例えばバックボーン等、インターネットの一部をなす回線）１５に接続され、同回線上のトラヒックを例えば一定時間毎にモニタしてセグメント（パケットと呼ばれることもある）を取得し、解析モジュール３に送る。

【００１５】

図１に示すネットワークでは、Ａ端末７はルータ１１とこのルータ１１に接続された２つのルータ１２、１３を介してインターネット１０に接続され、Ｂ端末８と端末９は共通のルータ１４を介してインターネット１０に接続され、セグメ

ント取得モジュール２はルータ１１、１２間でインターネット回線１５に接続されている。つまり、Ａ端末７とインターネット１０の接続ルートは２通り、ルータ１１とルータ１２を介したものと、ルータ１１とルータ１３を介してものがある。

【００１６】

今、図１のネットワークにおいて、Ａ端末７からＢ端末８へのトラヒックが実線１６ａで示すようにルータ１１からルータ１２、インターネット１０、ルータ１４を順に通るものであり、Ｂ端末８からＡ端末７へのトラヒックは実線１６ｂで示すようにルータ１４からインターネット１０、ルータ１３、ルータ１１を順に通るものである場合を考える。この場合、セグメント取得モジュール２は、Ａ端末７とＢ端末８間のトラヒックのうち、実線１６ａで示す一方向のトラヒック（Ａ端末７からＢ端末８へのトラヒック）のみしか取得できない。

また、Ｃ端末９からＡ端末７へのトラヒックが破線１７ａで示すようにルータ１４からインターネット１０、ルータ１２、ルータ１１を順に通るものであり、Ａ端末７からＣ端末９からへのトラヒックは破線１７ｂで示すようにルータ１１からルータ１３、インターネット１０、ルータ１４を順に通るものである場合を考える。この場合は、セグメント取得モジュール２は、Ｃ端末９とＡ端末７間のトラヒックのうち、破線１７ａで示す一方向のトラヒック（Ｃ端末９からＡ端末７へのトラヒック）のみしか取得できない。

【００１７】

解析モジュール３はセグメント取得モジュール３が取得したセグメントから、データ転送量等、トラヒック統計情報を計算する。

【００１８】

その際、解析モジュール３は、ＴＣＰヘッダ中のフラグビット６種類から、どの種類のセグメントを検出したかを識別する。

【００１９】

ここで、ＳＹＮビットのみ１となっているセグメントをＳＹＮセグメントと呼び、ＳＹＮビットとＡＣＫビットのみが共に１となっているセグメントをＳＹＮ＋ＡＣＫセグメントと呼ぶ。

【００２０】

また、ＳＹＮビット、ＦＩＮビット及びＡＣＫビットの３種類において、ＡＣＫビットのみが１となっているセグメントのうち、ユーザデータが存在するものをＤＡＴＡセグメントと呼び、ユーザデータが存在しないものをＡＣＫセグメントと呼ぶ。

【００２１】

更に、解析モジュール３は、取得したセグメントがどのコネクションに属するかを、例えば、取得したセグメントの発側ＩＰアドレス及び着側ＩＰアドレスに基づいて識別する。

【００２２】

また、解析モジュール３は、取得したセグメントがＤＡＴＡセグメントであればそのシーケンス番号、ユーザデータ長及び応答確認番号を認識し、取得したセグメントがＡＣＫセグメントであればその応答確認番号を認識する。

【００２３】

更に、解析モジュール３は、格納部５の状態遷移表を参照して、ＤＡＴＡセグメントを検出する毎に、今までに検出したＤＡＴＡセグメントのシーケンス番号から、次に送られてくるべきＤＡＴＡセグメントのシーケンス番号、つまり次シーケンス番号ｓｅｑ＿ｎｘｔを求めて管理する。この次シーケンス番号ｓｅｑ＿ｎｘｔは、今までに転送された最大のシーケンス番号と同じである。また、ＡＣＫセグメントを検出する毎に、今までに検出したＡＣＫセグメントの応答確認番号から、次に送られてくるべきＡＣＫセグメントの応答確認番号、つまり次応答確認番号ａｃｋ＿ｎｘｔを求めて管理し、また、最大の応答確認番号が検出された時のＡＣＫセグメントが持つウインドウサイズを、現ウインドウサイズｗｉｎとして求めて管理する。

【００２４】

蓄積モジュール４は、解析モジュール３の計算で得たトラヒック統計情報を、ＩＰアドレス別や、コネクション別、アプリケーション別に格納部６に格納する。

【００２５】

以下に、トラヒック統計情報収集の基礎となる解析モジュール３における計算例を説明する。ここで、取得したトラヒックがＳＹＮセグメント送信側のトラヒック（発側トラヒック）か、ＳＹＮ＋ＡＣＫセグメント送信側のトラヒック（着側トラヒック）かを区別することにより、クライアントからの通信か、サーバからの通信かを識別することができる。

【００２６】

［発側データ転送量の収集］

図２に示すシーケンスを参照して、発側データ転送量の収集を説明する。発側データ転送量としては発側転送セグメント数と発側転送バイト数があり、これらの収集には、モニタ可能な方向のトラヒックで、ＳＹＮセグメントを検出し、更に、このＳＹＮセグメントと同じコネクションに属するＤＡＴＡセグメントを検出する。

【００２７】

図２に示すように、モニタ可能な方向でＳＹＮセグメントが検出された場合、連続して検出されたＤＡＴＡセグメントの数を計算することにより、発側転送セグメント数を求める。

【００２８】

また、図２に示すように、ＳＹＮセグメントが検出された場合、連続して検出されたＤＡＴＡセグメントのうち、最初のＤＡＴＡセグメントのシーケンス番号ＳＥＱ１と、最後のＤＡＴＡセグメントのシーケンス番号ＳＥＱｎと、最後のＤＡＴＡセグメントのユーザデータ長ＬＥＮｎとを用い、式（１）を計算することにより、発側転送バイト数ｉｎｉ＿ｓｄｔを求める。

ｉｎｉ＿ｓｄｔ＝（ＳＥＱｎ＋ＬＥＮｎ）－ＳＥＱ１　…式（１）

即ち、発側転送バイト数ｉｎｉ＿ｓｄｔは、最初に検出したＤＡＴＡセグメントのシーケンス番号ＳＥＱ１と、最後に検出したＤＡＴＡセグメントのシーケンス番号ＳＥＱｎにそれのユーザデータ長ＬＥＮｎを加算した値（ＳＥＱｎ＋ＬＥＮｎ）との差である。

【００２９】

発側転送セグメント数と発側転送バイト数のうち、一方のみを発側データ転送

量として計算するようにしても良い。

【００３０】

[着側データ転送量の収集]

図３に示すシーケンスを参照して、着側データ転送量の収集を説明する。着側データ転送量としては着側転送セグメント数と着側転送バイト数があり、これらの収集には、モニタ可能な方向のトラヒックで、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、このＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＤＡＴＡセグメントを検出する。

【００３１】

図３に示すように、モニタ可能な方向でＳＹＮ＋ＡＣＫセグメントが検出された場合、連続して検出されたＤＡＴＡセグメントを計算することにより、着側転送セグメント数を求める。

【００３２】

また、図３に示すように、ＳＹＮ＋ＡＣＫセグメントが検出された場合、連続して検出されたＤＡＴＡセグメントのうち、最初のＤＡＴＡセグメントのシーケンス番号ＳＥＱ１と、最後のＤＡＴＡセグメントのシーケンス番号ＳＥＱｎと、そのユーザデータ長ＬＥＮｎとを用い、式（２）を計算して、着側転送バイト数ｒｓｐ＿ｓｄｔを求める。

ｒｓｐ＿ｓｄｔ＝（ＳＥＱｎ＋ＬＥＮｎ）－ＳＥＱ１　…式（２）

即ち、着側転送バイト数ｒｓｐ＿ｓｄｔは、最初に検出したＤＡＴＡセグメントのシーケンス番号ＳＥＱ１と、最後に検出したＤＡＴＡセグメントのシーケンス番号ＳＥＱｎにそれのユーザデータ長ＬＥＮｎを加算した値（ＳＥＱｎ＋ＬＥＮｎ）との差である。

【００３３】

この場合も、着側転送セグメント数と着側転送バイト数のうち、一方のみを着側データ転送量として計算するようにしても良い。

【００３４】

[発側データ受信量の収集]

図４に示すシーケンスを参照して、発側データ受信量（バイト数）の収集を説

明する。発側データ受信量の収集には、モニタ可能な方向のトラヒックで、ＳＹＮセグメントを検出し、更に、このＳＹＮセグメントと同じコネクションに属するＡＣＫセグメント及びＤＡＴＡセグメントを検出する。

【００３５】

図４に示すように、モニタ可能な方向でＳＹＮセグメントが検出された場合、連続して検出されたＡＣＫセグメントあるいはＤＡＴＡセグメントのうち、最初のＡＣＫセグメントあるいはＤＡＴＡセグメントの応答確認番号ＡＣＫ１と、最後のＡＣＫセグメントあるいはＤＡＴＡセグメントの応答確認番号ＡＣＫｎとを用い、式（３）を計算することにより、発側データ受信量ｉｎｉ＿ｒｄｔとする。

ｉｎｉ＿ｒｄｔ＝ＡＣＫｎ－ＡＣＫ１　…式（３）

即ち、発側受信バイト数ｉｎｉ＿ｒｄｔは、最初に検出したＡＣＫセグメントあるいはＤＡＴＡセグメントの応答確認番号ＡＣＫ１と、最後に検出したＡＣＫセグメントあるいはＤＡＴＡセグメントの応答確認番号ＡＣＫｎとの差である。

【００３６】

［着側データ受信量の収集］

図５に示すシーケンスを参照して、着側データ受信量（バイト数）の収集を説明する。着側データ受信量の収集には、モニタ可能な方向のトラヒックで、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、このＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＡＣＫセグメント及びＤＡＴＡセグメントを検出する。

【００３７】

図５に示すように、モニタ可能な方向でＳＹＮ＋ＡＣＫセグメントが検出された場合、連続して検出されたＡＣＫセグメントあるいはＤＡＴＡセグメントのうち、最初のＡＣＫセグメントあるいはＤＡＴＡセグメントの応答確認番号ＡＣＫ１と、最後のＡＣＫセグメントあるいはＤＡＴＡセグメントの応答確認番号ＡＣＫｎとを用い、式（４）を計算することにより、着側データ受信量ｒｓｐ＿ｒｄｔとする。

ｒｓｐ＿ｒｄｔ＝ＡＣＫｎ－ＡＣＫ１　…式（４）

即ち、着側受信バイト数ｒｓｐ＿ｒｄｔは、最初に検出したＡＣＫセグメントあ

るいはＤＡＴＡセグメントの応答確認番号ＡＣＫ１と、最後に検出したＡＣＫセグメントあるいはＤＡＴＡセグメントの応答確認番号ＡＣＫｎとの差である。

【００３８】

次に、図６を参照して、発側再送量及び着側再送量の収集を説明する。発側再送量及び着側再送量にはそれぞれ再送セグメント数と再送バイト数がある。

【００３９】

［発側再送量の収集］

発側再送量の収集には、モニタ可能な方向のトラヒックで、先ず、ＳＹＮセグメントを検出する（図示省略）。更に、このＳＹＮセグメントと同じコネクションに属するＤＡＴＡセグメントを検出する（図６（ａ）参照）。

【００４０】

モニタ可能な方向で、ＳＹＮセグメントに続いてＤＡＴＡセグメントが新たに検出された場合、この新たに検出したＤＡＴＡセグメントのシーケンス番号ＳＥＱが前述の別途管理により前回までのＤＡＴＡセグメントの検出で求めた次シーケンス番号ｓｅｑ＿ｎｘｔより小さいか否か判定し、この判定が肯定（ＳＥＱ＜ｓｅｑ＿ｎｘｔ）である場合に、ＳＹＮセグメントの送信側から再送があったと判断する。

【００４１】

発側再送量のうち、発側再送セグメント数ｉｎｉ＿ｒｅｔ＿ｎｕｍは、式（５）に示すように、今までの発側再送セグメント数に１を加算して求める。

ｉｎｉ＿ｒｅｔ＿ｎｕｍ＝ｉｎｉ＿ｒｅｔ＿ｎｕｍ＋１　…式（５）

【００４２】

発側再送バイト数ｉｎｉ＿ｒｅｔは、式（６）に示すように、重複のあった分、例えば図６（ｂ）にて斜線を付した部分１８、１９のバイト数を今までの発側再送バイト数に加算して求める。

ｉｎｉ＿ｒｅｔ＝ｉｎｉ＿ｒｅｔ

＋ｍｉｎ（ｓｅｑ＿ｎｘｔ－ＳＥＱ，ＬＥＮ）…式（６）

即ち、発側再送バイト数ｉｎｉ＿ｒｅｔは、次シーケンス番号ｓｅｑ＿ｎｘｔと新たに検出したＤＡＴＡセグメントのシーケンス番号ＳＥＱとの差（ｓｅｑ＿ｎ

ｘｔ－ＳＥＱ）と、この新たに検出したＤＡＴＡセグメントのユーザデータ長ＬＥＮとのうち、小さい方を今までの発側再送バイト数に加算した値である。

【００４３】

発側再送セグメント数と発側再送バイト数のうち、一方のみを発側再送量として計算するようにしても良い。

【００４４】

図６（ａ）に示す例では、２番目のＤＡＴＡセグメント（ＳＥＱ＝１４６０，ＬＥＮ＝１４６０）を検出した時点で、次シーケンス番号ｓｅｑ＿ｎｘｔは２９２０であるため、３番目のＤＡＴＡセグメント（ＳＥＱ＝０，ＬＥＮ＝１４６０）の検出により、再送があったと判断する。この時、ｓｅｑ＿ｎｘｔ－ＳＥＱ＝２９２０、ＬＥＮ＝１４６０であるから、図６（ｂ）に示す斜線部分（重複分）１８のバイト数は１４６０バイトであり、これが発側再送セグメント数とされる。

【００４５】

また、図６（ａ）に示す例では、４番目のＤＡＴＡセグメント（ＳＥＱ＝２９２０，ＬＥＮ＝１４６０）を検出した時点で、次シーケンス番号ｓｅｑ＿ｎｘｔは４３８０であるため、５番目のＤＡＴＡセグメント（ＳＥＱ＝４０００，ＬＥＮ＝１４６０）の検出により、再送があったと判断する。この時、ｓｅｑ＿ｎｘｔ－ＳＥＱ＝３８０、ＬＥＮ＝１４６０であるから、図６（ｂ）に示す斜線部分（重複分）１９のバイト数は３８０バイトであり、これを今までの発側再送セグメント数１４６０に加算する。従って、新たな発側再送セグメント数は１８４０となる。

【００４６】

発側再送セグメント数と発側再送バイト数のうち、一方のみを発側再送量として計算するようにしても良い。

【００４７】

[着側再送量の収集]

着側再送量の収集には、モニタ可能な方向のトラヒックで、先ず、ＳＹＮ＋ＡＣＫセグメントを検出する（図示省略）。更に、このＳＹＮ＋ＡＣＫセグメント

と同じコネクションに属するＤＡＴＡセグメントを検出する（図６（ａ）参照）。

【００４８】

モニタ可能な方向で、ＳＹＮ＋ＡＣＫセグメントに続いてＤＡＴＡセグメントが新たに検出された場合、この新たに検出したＤＡＴＡセグメントのシーケンス番号ＳＥＱが前述の別途管理により前回までのＤＡＴＡセグメントの検出で求めた次シーケンス番号ｓｅｑ＿ｎｘｔより小さいか否か判定し、この判定が肯定（ＳＥＱ＜ｓｅｑ＿ｎｘｔ）である場合に、ＳＹＮ＋ＡＣＫセグメントの送信側から再送があったと判断する。

【００４９】

着側再送量のうち、着側再送セグメント数ｒｓｐ＿ｒｅｔ＿ｎｕｍは、式（７）に示すように、今までの着側再送セグメント数に１を加算して求める。

ｒｓｐ＿ｒｅｔ＿ｎｕｍ＝ｒｓｐ＿ｒｅｔ＿ｎｕｍ＋１　…式（７）

【００５０】

着側再送バイト数ｒｓｐ＿ｒｅｔは、式（８）に示すように、重複のあった分、例えば図６（ｂ）にて斜線を付した部分１８、１９のバイト数を今までの着側再送バイト数に加算して求める。

ｒｓｐ＿ｒｅｔ＝ｒｓｐ＿ｒｅｔ
＋ｍｉｎ（ｓｅｑ＿ｎｘｔ－ＳＥＱ，ＬＥＮ）…式（８）

即ち、着側再送バイト数ｒｓｐ＿ｒｅｔは、次シーケンス番号ｓｅｑ＿ｎｘｔと新たに検出したＤＡＴＡセグメントのシーケンス番号ＳＥＱとの差（ｓｅｑ＿ｎｘｔ－ＳＥＱ）と、この新たに検出したＤＡＴＡセグメントのユーザデータ長ＬＥＮとのうち、小さい方を今までの着側再送バイト数に加算した値である。

【００５１】

この場合も、着側再送セグメント数と着側再送バイト数のうち、一方のみを着側再送量として計算するようにしても良い。

【００５２】

次に、図７を参照して、発側ＤＡＴＡセグメント紛失数及び着側ＤＡＴＡセグメント紛失数の収集を説明する。

【００５３】

［発側ＤＡＴＡセグメント紛失数の収集］

発側ＤＡＴＡセグメント紛失数の収集には、モニタ可能な方向のトラヒックで、先ず、ＳＹＮセグメントを検出する（図示省略）。更に、このＳＹＮセグメントと同じコネクションに属するＡＣＫセグメントを検出する（図７参照）。

【００５４】

モニタ可能な方向で、ＳＹＮセグメントに続いてＡＣＫセグメントが新たに検出された場合、この新たに検出したＡＣＫセグメントの応答確認番号ＡＣＫ及びウインドウサイズＷＩＮを、前述の別途管理している次応答確認番号ａｃｋ＿ｎｘｔ及び現ウィンドウサイズｗｉｎとそれぞれ比較し、式（９）と式（１０）の両方が成立するとき、ＤＡＴＡセグメント紛失があったと判断する。

ＡＣＫ＝ａｃｋ＿ｎｘｔ　…式（９）

ＷＩＮ＝ｗｉｎ　…式（１０）

【００５５】

但し、応答確認番号ＡＣＫとウインドウサイズＷＩＮの組が同じ値のＡＣＫセグメントについて２回以上ＤＡＴＡセグメント紛失があったと判断した場合のみ、式（１１）に示すように、今までの発側ＤＡＴＡセグメント紛失数に１を加算して、新たな発側ＤＡＴＡセグメント紛失数ｉｎｉ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍとする。

ｉｎｉ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍ＝ｉｎｉ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍ＋１　…式（１１）

【００５６】

このような計算（応答確認番号ＡＣＫ及びウインドウサイズＷＩＮの組が同じ値の２つ以上のＡＣＫセグメントについて式（９）と式（１０）の両方が成立する場合のみ、ＤＡＴＡセグメント紛失数に１だけ加算）を行うため、本例では、重複ＡＣＫフラグ（ｄｕｐａｃｋと表記する）を導入している。

【００５７】

そして、式（１２）に示すように重複ＡＣＫフラグ（ｄｕｐａｃｋ）が０である場合のみ、式（１３）及び式（１４）の両方が成立するか否か判断する。但し

、式（１３）及び式（１４）は、前の式（９）及び式（１０）と同じである。

ｄｕｐａｃｋ＝０　　…式（１２）

ＡＣＫ＝ａｃｋ＿ｎｘｔ　…式（１３）

ＷＩＮ＝ｗｉｎ　　…式（１４）

【００５８】

更に、前式（１２）、前式（１３）及び前式（１４）が全て成立する場合のみ、式（１５）により発側ＤＡＴＡセグメント紛失数ｉｎｉ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍに１を加算し、且つ、式（１６）のように重複ＡＣＫフラグ（ｄｕｐａｃｋ）を１に変更するようにしている。但し、式（１５）は前の式（１１）と同じである。

ｉｎｉ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍ＝ｉｎｉ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍ＋１　　…式（１５）

ｄｕｐａｃｋ＝０　　…式（１６）

【００５９】

ここで、重複ＡＣＫフラグ（ｄｕｐａｃｋ）の値は、解析モジュール３が下記のように管理する。

(1) 重複ＡＣＫフラグの初期値を０とする。

(2) 初期状態（ｄｕｐａｃｋ＝０）又はｄｕｐａｃｋ＝０にて、前述の式（１３）と式（１４）の両方が成立するＡＣＫセグメントを始めて検出した時点で、重複ＡＣＫフラグの値を０から１に変更する（ｄｕｐａｃｋ＝１）。

(3) ｄｕｐａｃｋ＝１の状態で、重複ＡＣＫフラグの値を０から１に変更した時のＡＣＫセグメントと応答確認番号の値あるいはウインドウサイズの値が異なるＡＣＫセグメントを検出した時点で、重複ＡＣＫフラグの値を１から０に戻す（ｄｕｐａｃｋ＝０）。

(4) ｄｕｐａｃｋ＝１の状態では、重複ＡＣＫフラグの値を０から１に変更した時のＡＣＫセグメントと同じ値の応答確認番号とウインドウサイズの組を持つＡＣＫセグメントを次に検出しても、重複ＡＣＫフラグの値はｄｕｐａｃｋ＝１のままで変更しない。

【００６０】

以上のように、ＳＹＮセグメントを検出し、更に、検出したＳＹＮセグメントと同じコネクションに属するＡＣＫセグメントを検出した場合、新たに検出したＡＣＫセグメントの応答確認番号ＡＣＫ及びウインドウサイズＷＩＮの両方が次応答確認番号ａｃｋ＿ｎｘｔ及び現ウインドウサイズｗｉｎに等しいか否かを判定し、応答確認番号ＡＣＫ及びウインドウサイズＷＩＮの組が同じ値の２つ以上のＡＣＫセグメントについて前述の式（９）と式（１０）の判定が共に肯定である毎に、今までの発側ＤＡＴＡセグメント紛失数に１を加算して新たな発側ＤＡＴＡセグメント紛失数ｉｎｉ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍを求める。

【００６１】

図７に示すシーケンスにおいて、１番目のＡＣＫセグメントの応答確認番号はＡＣＫ＝１４６０、ＷＩＮ＝８１９２である。従って、最初は、次応答確認番号はａｃｋ＿ｎｘｔ＝１４６０、現ウインドウサイズはｗｉｎ＝８１９２である。重複ＡＣＫフラグ（ｄｕｐａｃｋ）は初期値０のままである。

【００６２】

２番目のＡＣＫセグメントの応答確認番号はＡＣＫ＝２９２０、そのウインドウサイズはＷＩＮ＝８１９２であることから、ＡＣＫ≠ａｃｋ＿ｎｘｔである（応答確認番号ＡＣＫ＝２９２０が次応答確認番号ａｃｋ＿ｎｘｔ＝１４６０に等しくない）ため、次応答確認番号はａｃｋ＿ｎｘｔ＝２９２０となり、重複ＡＣＫフラグ（ｄｕｐａｃｋ）は０のままである。

【００６３】

３番目のＡＣＫセグメントの応答確認番号とウインドウサイズの組（ＡＣＫ＝２９２０，ＷＩＮ＝８１９２）は２番目のＡＣＫセグメントと同じ値であり、それぞれ次応答確認番号ａｃｋ＿ｎｘｔ（＝２９２０）と現ウインドウサイズｗｉｎ（＝８１９２）に一致する。従って、３番目のＡＣＫセグメントを検出した時点で、発側ＤＡＴＡセグメント紛失数ｉｎｉ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍは、今までの０に１が加算されて、１となる。また、重複ＡＣＫフラグは０から１へ変わる。

【００６４】

４番目のＡＣＫセグメントの応答確認番号とウインドウサイズの組（ＡＣＫ＝

２９２０，ＷＩＮ＝８１９２）も２番目のＡＣＫセグメントと同じ値であり、それぞれ次応答確認番号ａｃｋ＿ｎｘｔ（＝２９２０）と現ウインドウサイズｗｉｎ（＝８１９２）に一致する。しかし、この時点までは重複ＡＣＫフラグは１であるから、発側ＤＡＴＡセグメント紛失数の加算はない。また、重複ＡＣＫフラグは１のままである。

【００６５】

５番目のＡＣＫセグメントの応答確認番号はＡＣＫ＝４３８０であり、そのウインドウサイズはＷＩＮ＝８１９２であり、２～４番目のＡＣＫセグメント（ＡＣＫ＝２９２０，ＷＩＮ＝８１９２）とは応答確認番号とウインドウサイズの組、特に応答確認番号が異なる。

【００６６】

一方、４番目のＡＣＫセグメントを検出した時点では、次応答確認番号はａｃｋ＿ｎｘｔ＝２９２０であり、現ウインドウサイズはｗｉｎ＝８１９２である。

【００６７】

従って、５番目のＡＣＫセグメントを検出した時点で、ＡＣＫ≠ａｃｋ＿ｎｘｔである（応答確認番号ＡＣＫ＝４３８０が次応答確認番号ａｃｋ＿ｎｘｔ＝２９２０に等しくない）から、次応答確認番号はａｃｋ＿ｎｘｔ＝４３８０となり、重複ＡＣＫフラグ（ｄｕｐａｃｋ）は０に戻る。

【００６８】

６番目のＡＣＫセグメントの応答確認番号とウインドウサイズの組（ＡＣＫ＝４３８０，ＷＩＮ＝８１９２）は５番目のＡＣＫセグメントと同じ値であり、それぞれ次応答確認番号ａｃｋ＿ｎｘｔ（＝４３８０）と現ウインドウサイズｗｉｎ（＝８１９２）に一致する。従って、６番目のＡＣＫセグメントを検出した時点で、発側ＤＡＴＡセグメント紛失数ｉｎｉ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍは、今までの１に１が加算されて、２となる。また、重複ＡＣＫフラグは０から１へ変わる。

【００６９】

７番目と８番目のＡＣＫセグメントを検出しても、発側ＤＡＴＡセグメント紛失数の加算はない。即ち、

(1) ７番目のＡＣＫセグメントの応答確認番号はＡＣＫ＝５８４０であり、そのウインドウサイズはＷＩＮ＝８１９２であり、６番目のＡＣＫセグメント（ＡＣＫ＝４３８０，ＷＩＮ＝８１９２）とは応答確認番号とウインドウサイズの組、特に応答確認番号が異なる。一方、６番目のＡＣＫセグメントを検出した時点では、次応答確認番号はａｃｋ＿ｎｘｔ＝４３８０であり、現ウインドウサイズはｗｉｎ＝８１９２である。従って、７番目のＡＣＫセグメントを検出した時点で、ＡＣＫ≠ａｃｋ＿ｎｘｔである（応答確認番号ＡＣＫ＝５８４０が次応答確認番号ａｃｋ＿ｎｘｔ＝４３８０に等しくない）から、次応答確認番号はａｃｋ＿ｎｘｔ＝５８４０となり、重複ＡＣＫフラグ（ｄｕｐａｃｋ）は０に戻る。

(2) ８番目のＡＣＫセグメントの応答確認番号はＡＣＫ＝５８４０であり、そのウインドウサイズはＷＩＮ＝１６３８４であり、７番目のＡＣＫセグメント（ＡＣＫ＝５８４０，ＷＩＮ＝８１９２）とは応答確認番号とウインドウサイズの組、特にウインドウサイズが異なる。一方、７番目のＡＣＫセグメントを検出した時点では、次応答確認番号はａｃｋ＿ｎｘｔ＝５８４０であり、現ウインドウサイズはｗｉｎ＝８１９２である。従って、８番目のＡＣＫセグメントを検出した時点で、ＷＩＮ≠ｗｉｎである（ウインドウサイズＷＩＮ＝１６３８４が現ウインドウサイズｗｉｎ＝８１９２に等しくない）から、現ウインドウサイズはｗｉｎ＝１６３８４となり、重複ＡＣＫフラグ（ｄｕｐａｃｋ）は０のままである。

【００７０】

［着側ＤＡＴＡセグメント紛失数の収集］

着側ＤＡＴＡセグメント紛失数の収集には、モニタ可能な方向のトラヒックで、先ず、ＳＹＮ＋ＡＣＫセグメントを検出する（図示省略）。更に、このＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＡＣＫセグメントを検出する（図７参照）。

【００７１】

モニタ可能な方向で、ＳＹＮ＋ＡＣＫセグメントに続いてＡＣＫセグメントが新たに検出された場合、この新たに検出したＡＣＫセグメントの応答確認番号ＡＣＫ及びウインドウサイズＷＩＮを、前述の別途管理している次応答確認番号ａｃｋ＿ｎｘｔ及び現ウィンドウサイズｗｉｎとそれぞれ比較する。

【００７２】

そして、先に説明した発側ＤＡＴＡセグメント紛失数の収集と同様、重複ＡＣＫフラグ（ｄｕｐａｃｋ）を導入し、式（１７）、式（１８）及び式（１９）の全てが成立するか否か判断する。

ｄｕｐａｃｋ＝０　　…式（１７）

ＡＣＫ＝ａｃｋ＿ｎｘｔ　…式（１８）

ＷＩＮ＝ｗｉｎ　　…式（１９）

【００７３】

式（１７）に示したように重複ＡＣＫフラグが０の場合に、前記の式（１８）と式（１９）の両方が成立すれば、式（２０）に示すように着側ＤＡＴＡセグメント紛失数ｒｓｐ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍに１を加算する。つまり、応答確認番号ＡＣＫとウインドウサイズＷＩＮの組が同じ値のＡＣＫセグメントについて２回以上ＤＡＴＡセグメント紛失があったと判断した場合のみ、式（２０）に示すように、今までの着側ＤＡＴＡセグメント紛失数に１を加算して、新たな着側ＤＡＴＡセグメント紛失数ｒｓｐ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍとする。また、式（２１）に示すように重複ＡＣＫフラグを１に変更する。

ｒｓｐ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍ＝ｒｓｐ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍ

＋１　　…式（２０）

ｄｕｐａｃｋ＝１　　…式（２１）

【００７４】

言い換えれば、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、検出したＳＹＮ＋ＡＣＫセグメントと同じコネクションに属するＡＣＫセグメントを検出した場合、新たに検出したＡＣＫセグメントの応答確認番号ＡＣＫ及びウインドウサイズＷＩＮの両方が次応答確認番号ａｃｋ＿ｎｘｔ及び現ウインドウサイズｗｉｎに等しいか否かを判定し、応答確認番号ＡＣＫ及びウインドウサイズＷＩＮの組が同じ値の２つ以上のＡＣＫセグメントについて前述の式（１８）と式（１９）の判定が共に肯定である毎に、今までの着側ＤＡＴＡセグメント紛失数に１を加算して新たな着側ＤＡＴＡセグメント紛失数ｒｓｐ＿ｒｅｃｖ＿ｄｒｏｐ＿ｎｕｍを求める。

【００７５】

次に、図８（ａ）、（ｂ）を参照して、ＨＴＴＰの応答時間及びスループットの収集を説明する。

【００７６】

［ＨＴＴＰ通信の説明］

ＨＴＴＰの通信は、クライアントからの要求を表すＨＴＴＰリクエストと、それに対応するサーバからの応答を表すＨＴＴＰレスポンスにより構成される。ＨＴＴＰリクエスト及びＨＴＴＰレスポンスはそれぞれ複数のＤＡＴＡセグメントで構成され、通常は、テキストや画像等のコンテンツを含むＨＴＴＰレスポンスの方が、ＤＡＴＡセグメントの総バイト数が大きくなる。また、クライアント及びサーバからの各ＤＡＴＡセグメント群に対応して、複数のＡＣＫセグメントがサーバ及びクライアントから送られる。

【００７７】

［ＨＴＴＰの発側応答時間及び発側スループットの収集］

図８（ａ）はＨＴＴＰの発側応答時間及び発側スループットの収集を説明するためのシーケンス図であり、これらの収集には、モニタ可能な方向のトラヒックで、ＳＹＮセグメントを検出し、更に、このＳＹＮセグメントと同じコネクションに属する連続複数のＤＡＴＡセグメント（ＨＴＴＰリクエスト）及びこのＨＴＴＰリクエストに続く連続複数のＡＣＫセグメント（ＨＴＴＰレスポンス）を検出する。

【００７８】

ＨＴＴＰ発側応答時間ｉｎｉ＿ｈｔｔｐ＿ｒｓｐ＿ｔｉｍｅは、ＨＴＴＰリクエストの最後のセグメントとＨＴＴＰレスポンスの最初のセグメントとの時間差を計算することにより、求まる。ＨＴＴＰリクエストとＨＴＴＰレスポンスとの区切りは、連続して転送されるＤＡＴＡセグメントとＡＣＫセグメントとの切り替わりによって検出できる。

【００７９】

従って、図８（ａ）に示すようにモニタ可能な方向でＳＹＮセグメントが検出された場合、最後のＤＡＴＡセグメントの検出から最初のＡＣＫセグメントの検

出までの時刻差を、ＨＴＴＰ発側応答時間ｉｎｉ＿ｈｔｔｐ＿ｒｓｐ＿ｔｉｍｅとして計算する。

【００８０】

一方、ＨＴＴＰ発側スループットｉｎｉ＿ｈｔｔｐ＿ｔｐｔは、ＨＴＴＰレスポンスの総バイト数と、ＨＴＴＰレスポンスの最初のセグメントから最後のセグメントまでの時刻差との比とする。これは、通常ＨＴＴＰリクエストよりもＨＴＴＰレスポンスの方がＤＡＴＡセグメントの総バイト数が大きいので、精度の点で、ＨＴＴＰレスポンスを転送している間のスループットを求めることが好ましいからである。

【００８１】

そこで、図８（ａ）に示すようにモニタ可能な方向でＳＹＮセグメントが検出された場合、ＨＴＴＰレスポンスの総バイト数ｈｔｔｐ＿ｒｓｐ＿ｂｙｔを、最初に検出したＡＣＫセグメントと最後に検出したＡＣＫセグメントとの応答確認番号の差を計算して求める。また、ＨＴＴＰレスポンスの最初のセグメントから最後のセグメントまでの時刻差ｈｔｔｐ＿ｒｓｐ＿ｄｔｉｍｅを、最初のＡＣＫセグメントの検出から最後のＡＣＫセグメントの検出までの時刻差を計算して求める。

【００８２】

更に、式（２２）に示すように、総バイト数ｈｔｔｐ＿ｒｓｐ＿ｂｙｔを時刻差ｈｔｔｐ＿ｒｓｐ＿ｄｔｉｍｅで割算することにより、ＨＴＴＰ発側スループットｉｎｉ＿ｈｔｔｐ＿ｔｐｔを求める。

ｉｎｉ＿ｈｔｔｐ＿ｔｐｔ＝ｈｔｔｐ＿ｒｓｐ＿ｂｙｔ／ｈｔｔｐ＿ｒｓｐ＿ｄｔｉｍｅ　…式（２２）

【００８３】

但し、ＨＴＴＰ発側スループットｉｎｉ＿ｈｔｔｐ＿ｔｐｔの計算において、異常値を採らないように、連続して検出されたＡＣＫセグメントの数が所定の閾値（１０程度の定数）以上の場合のみ、スループットとして算出している。

【００８４】

ＨＴＴＰ発側応答時間とＨＴＴＰ発側スループットのうち、一方のみを計算す

るようにしても良い。また、ＨＴＴＰリクエストの総バイト数と、ＨＴＴＰリクエストの最初のセグメントから最後のセグメントまでの時刻差との比を計算して、ＨＴＴＰ発側スループットを求めることも可能である。

【００８５】

［ＨＴＴＰの着側応答時間及び着側スループットの収集］

図８（ｂ）はＨＴＴＰの着側応答時間及び着側スループットの収集を説明するためのシーケンス図であり、これらの収集には、モニタ可能な方向のトラヒックで、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、このＳＹＮ＋ＡＣＫセグメントと同じコネクションに属する連続複数のＡＣＫセグメント（ＨＴＴＰリクエスト）及びこのＨＴＴＰリクエストに続く連続複数のＤＡＴＡセグメント（ＨＴＴＰレスポンス）を検出する。

【００８６】

ＨＴＴＰ着側応答時間ｒｓｐ＿ｈｔｔｐ＿ｒｓｐ＿ｔｉｍｅも、ＨＴＴＰリクエストの最後のセグメントとＨＴＴＰレスポンスの最初のセグメントとの時間差を計算することにより求まる。ＨＴＴＰリクエストとＨＴＴＰレスポンスとの区切りは、連続して転送されるＡＣＫセグメントとＤＡＴＡセグメントとの切り替わりによって検出できる。

【００８７】

そこで、図８（ｂ）に示すようにモニタ可能な方向でＳＹＮ＋ＡＣＫセグメントが検出された場合、最後のＡＣＫセグメントの検出から最初のＤＡＴＡセグメントの検出までの時刻差を、ＨＴＴＰ着側応答時間ｒｓｐ＿ｈｔｔｐ＿ｒｓｐ＿ｔｉｍｅとして計算する。

【００８８】

ＨＴＴＰ着側スループットｒｓｐ＿ｈｔｔｐ＿ｔｐｔは、ＨＴＴＰレスポンスの総バイト数と、ＨＴＴＰレスポンスの最初のセグメントから最後のセグメントまでの時刻差との比とする。これも、通常ＨＴＴＰリクエストよりもＨＴＴＰレスポンスの方がＤＡＴＡセグメントの総バイト数が大きいので、精度の点で、ＨＴＴＰレスポンスを転送している間のスループットを求めることが好ましいからである。

【００８９】

そこで、図８（ｂ）に示すようにモニタ可能な方向でＳＹＮ＋ＡＣＫセグメントが検出された場合、ＨＴＴＰレスポンスの総バイト数ｈｔｔｐ＿ｒｓｐ＿ｂｙｔを、最初に検出したＤＡＴＡセグメントのシーケンス番号と最後に検出したＤＡＴＡセグメントのシーケンス番号にそれのユーザデータ長を加算した値との差を計算して求める。また、ＨＴＴＰレスポンスの最初のセグメントから最後のセグメントまでの時刻差ｈｔｔｐ＿ｒｓｐ＿ｄｔｉｍｅを、最初のＤＡＴＡセグメントの検出から最後のＤＡＴＡセグメントの検出までの時刻差を計算して求める。

【００９０】

更に、式（２３）に示すように、総バイト数ｈｔｔｐ＿ｒｓｐ＿ｂｙｔを時刻差ｈｔｔｐ＿ｒｓｐ＿ｄｔｉｍｅで割算することにより、ＨＴＴＰ着側スループットｒｓｐ＿ｈｔｔｐ＿ｔｐｔを求める。

ｒｓｐ＿ｈｔｔｐ＿ｔｐｔ＝ｈｔｔｐ＿ｒｓｐ＿ｂｙｔ／ｈｔｔｐ＿ｒｓｐ＿ｄｔｉｍｅ　…式（２３）

【００９１】

ＨＴＴＰ着側スループットｒｓｐ＿ｈｔｔｐ＿ｔｐｔの計算において、異常値を採らないように、連続して検出されたＡＣＫセグメントの数が所定の閾値（１０程度の定数）以上の場合のみ、スループットとして算出している。

【００９２】

この場合も、ＨＴＴＰ着側応答時間とＨＴＴＰ着側スループットのうち、一方のみを計算するようにしても良い。また、ＨＴＴＰリクエストの総バイト数と、ＨＴＴＰリクエストの最初のセグメントから最後のセグメントまでの時刻差との比を計算して、ＨＴＴＰ着側スループットを求めることも可能である。

【００９３】

次に、図９（ａ）、（ｂ）を参照して、ＦＴＰのスループットの収集を説明する。

【００９４】

［ＦＴＰ通信の説明］

ＦＴＰの通信は、制御データを転送するための制御用ＴＣＰコネクションと、ファイルを転送するためのデータ用ＴＣＰコネクションを用いて行われる。このうち、データ用ＴＣＰコネクションでは、サーバからクライアントへ大量のバイト数（大量のＤＡＴＡセグメント）が転送される可能性があるため、スループットの計算を行うことが可能である。サーバからの大量のＤＡＴＡセグメントに対応して、クライアントからサーバへＡＣＫセグメントが大量に送られる。但し、前述のように、データ用ＴＣＰコネクションではサーバがＳＹＮセグメントを送信し、クライアントがＳＹＮ＋ＡＣＫセグメントを送信する。従って、ＦＴＰのスループットを収集する場合、発側とはＳＹＮ＋ＡＣＫセグメントの送信側となり、着側とはＳＹＮセグメントの送信側となる。

【００９５】

［ＦＴＰの発側スループットの収集］

図９（ａ）はＦＴＰの発側スループットの収集を説明するためのシーケンス図であり、これの収集には、モニタ可能な方向のトラヒックで、ＳＹＮ＋ＡＣＫセグメントを検出し、更に、このＳＹＮ＋ＡＣＫセグメントと同じコネクションに属する連続複数のＡＣＫセグメントを検出する。

【００９６】

ＦＴＰ発側スループットｉｎｉ＿ｆｔｐ＿ｔｐｔは、ファイルの総バイト数と、ファイル転送の最初のセグメントから最後のセグメントまでの時刻差との比とする。

【００９７】

そこで、図９（ａ）に示すようにモニタ可能な方向でＳＹＮ＋ＡＣＫセグメントが検出された場合、ファイルの総バイト数ｆｔｐ＿ｂｙｔを、最初に検出したＡＣＫセグメントと最後に検出したＡＣＫセグメントとの応答確認番号の差を計算して求める。また、ファイル転送の最初のセグメントから最後のセグメントまでの時刻差ｆｔｐ＿ｄｔｉｍｅを、最初のＡＣＫセグメントの検出から最後のＡＣＫセグメントの検出までの時刻差を計算して求める。

【００９８】

更に、式（２４）に示すように、総バイト数ｆｔｐ＿ｂｙｔを時刻差ｆｔｐ＿

ｄｔｉｍｅで割算することにより、ＦＴＰ発側スループットｉｎｉ＿ｆｔｐ＿ｔｐｔを求める。

ｉｎｉ＿ｆｔｐ＿ｔｐｔ＝ｆｔｐ＿ｂｙｔ／ｆｔｐ＿ｄｔｉｍｅ

…式（２４）

【００９９】

但し、ＦＴＰ発側スループットｉｎｉ＿ｆｔｐ＿ｔｐｔの計算において、異常値を採らないように、連続して検出されたＡＣＫセグメントの数が所定の閾値（１０程度の定数）以上の場合のみ、スループットとして算出している。

【０１００】

［ＦＴＰの着側スループットの収集］

図９（ｂ）はＦＴＰの着側スループットの収集を説明するためのシーケンス図であり、これの収集には、モニタ可能な方向のトラヒックで、ＳＹＮセグメントを検出し、更に、このＳＹＮセグメントと同じコネクションに属する連続複数のＤＡＴＡセグメントを検出する。

【０１０１】

ＦＴＰ着側スループットｒｓｐ＿ｆｔｐ＿ｔｐｔも、ファイルの総バイト数と、ファイル転送の最初のセグメントから最後のセグメントまでの時刻差との比とする。

【０１０２】

そこで、図９（ｂ）に示すようにモニタ可能な方向でＳＹＮセグメントが検出された場合、ファイルの総バイト数ｆｔｐ＿ｂｙｔを、最初に検出したＤＡＴＡセグメントのシーケンス番号と最後に検出したＤＡＴＡセグメントのシーケンス番号にそれのユーザデータ長を加算した値との差を計算して求める。また、ファイル転送の最初のセグメントから最後のセグメントまでの時刻差ｆｔｐ＿ｄｔｉｍｅを、最初のＤＡＴＡセグメントの検出から最後のＤＡＴＡセグメントの検出までの時刻差を計算して求める。

【０１０３】

更に、式（２５）に示すように、総バイト数ｆｔｐ＿ｂｙｔを時刻差ｆｔｐ＿ｄｔｉｍｅで割算することにより、ＦＴＰ着側スループットｒｓｐ＿ｆｔｐ＿ｔ

ｐｔを求める。

ｒｓｐ＿ｆｔｐ＿ｔｐｔ＝ｆｔｐ＿ｂｙｔ／ｆｔｐ＿ｄｔｉｍｅ

…式（２５）

【０１０４】

但し、ＦＴＰ着側スループットｒｓｐ＿ｆｔｐ＿ｔｐｔの計算においても、異常値を採らないように、連続して検出されたＤＡＴＡセグメントの数が所定の閾値（１０程度の定数）以上の場合のみ、スループットとして算出している。

【０１０５】

以上のように、各種の情報収集において、ＳＹＮセグメントを検出したかＳＹＮ＋ＡＣＫセグメントを検出したかによって発側のトラヒック統計情報か着側のトラヒック統計情報かを区別することにより、つまり、モニタした方向がＳＹＮセグメントの送信側かＳＹＮ＋ＡＣＫセグメントの送信側かを区別することにより、クライアントからの通信かサーバからの通信かを識別することができる。

【０１０６】

【発明の効果】

以上説明したように、本発明によれば、片方向のトラヒックしか取得できない場合においても、反対方向からのトラヒックが推定可能であり、より多くのトラヒック統計情報を収集することができる。従って、本発明はインターネットのバックボーン回線のトラヒック調査に極めて有用である。

【図面の簡単な説明】

【図１】

本発明の実施形態例として、双方向トラヒック統計情報収集装置の構成例を示す図。

【図２】

本発明の実施形態例として、発側データ転送量を収集する場合のシーケンス例を示す図。

【図３】

本発明の実施形態例として、着側データ転送量を収集する場合のシーケンス例を示す図。

【図４】

本発明の実施形態例として、発側データ受信量を収集する場合のシーケンス例を示す図。

【図５】

本発明の実施形態例として、着側データ受信量を収集する場合のシーケンス例を示す図。

【図６】

本発明の実施形態例として、データ受信量を収集する場合のシーケンス例、並びに、重複バイトの例を示す図。

【図７】

本発明の実施形態例として、セグメント紛失数を収集する場合のシーケンス例を示す図。

【図８】

本発明の実施形態例として、ＨＴＴＰの応答時間及びスループットを収集する場合のシーケンス例を示す図。

【図９】

本発明の実施形態例として、ＦＴＰのスループットを収集する場合のシーケンス例を示す図。

【符号の説明】

１　双方向トラヒック統計情報収集装置

２　セグメント取得モジュール

３　解析モジュール

４　蓄積モジュール

５　状態遷移表の格納部

６　トラヒック統計情報の格納部

７、８、９　端末

１０　インターネット

１１、１２、１３、１４　ルータ

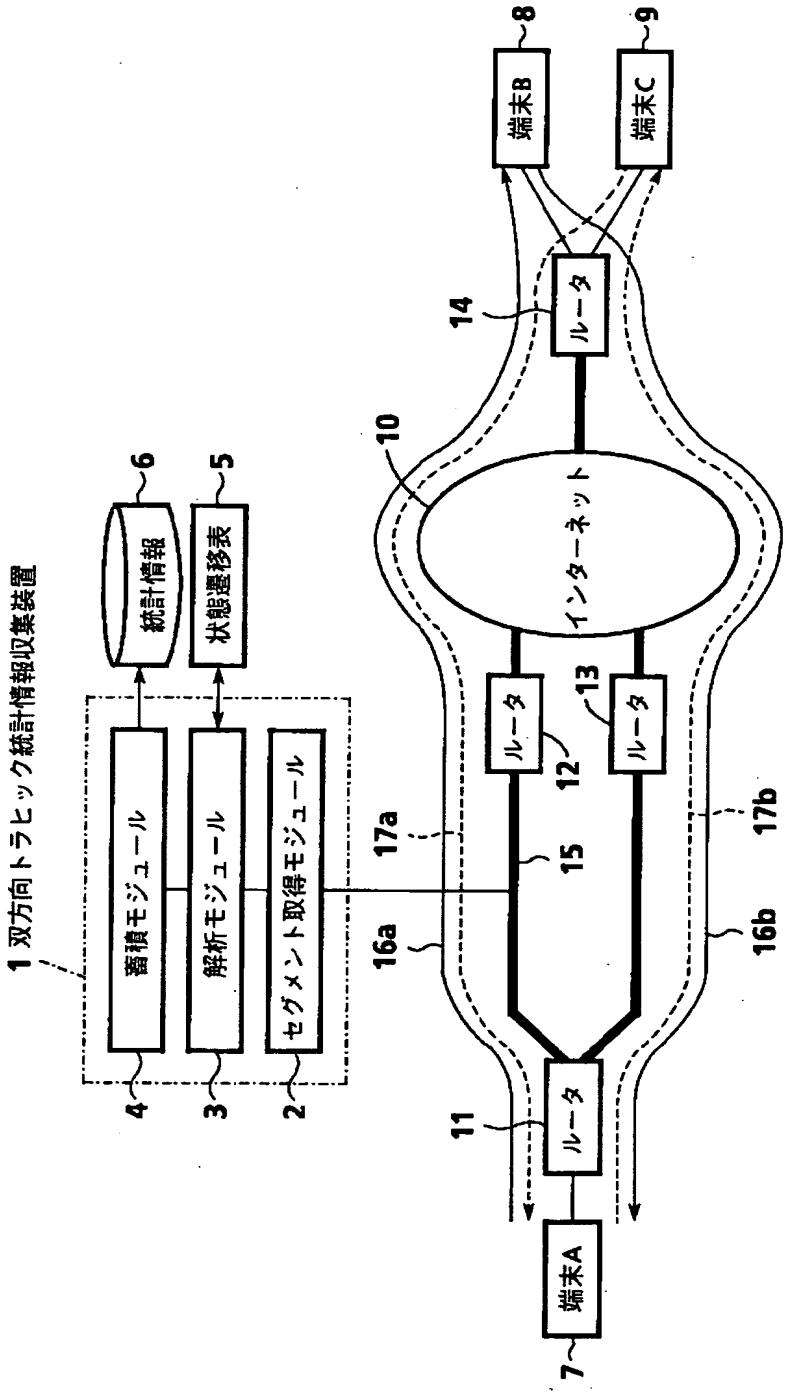
１５　バックボーン

１６ａ、１７ａ　セグメントの取得可能なトラヒック

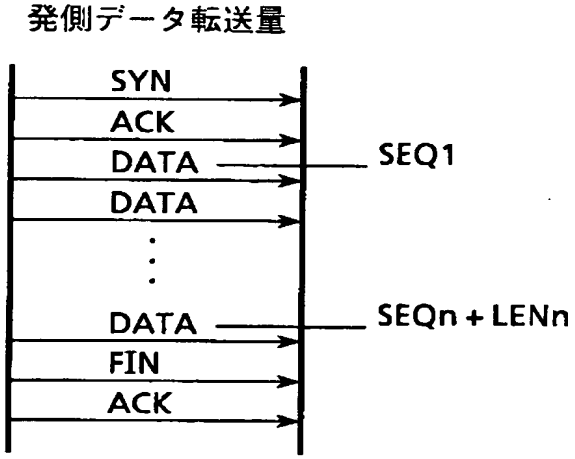
１８、１９　重複バイト部分

【書類名】　　図面

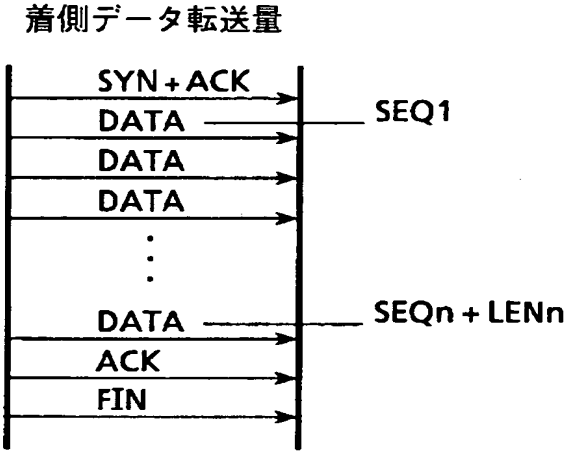
【図１】


1 双方向トラヒック統計情報収集装置
蓄積モジュール
解析モジュール
セグメント取得モジュール
統計情報
状態遷移表
2
3
4
5
6
インターネット
10
ルータ
11
12
13
14
15
16a
16b
17a
17b
端末A
端末B
端末C
7
8
9

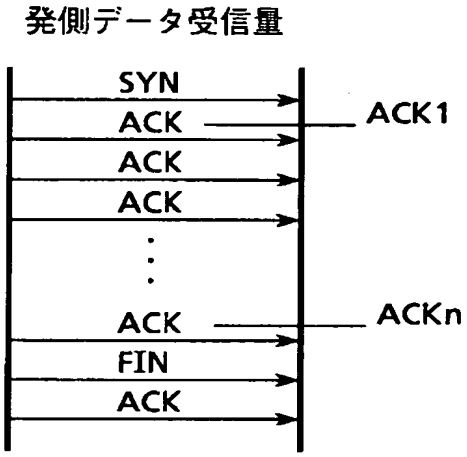
【図２】



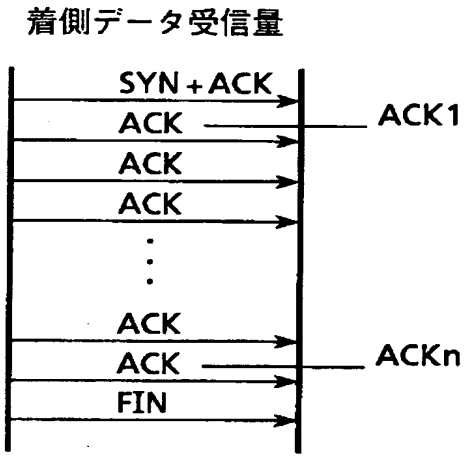
【図３】



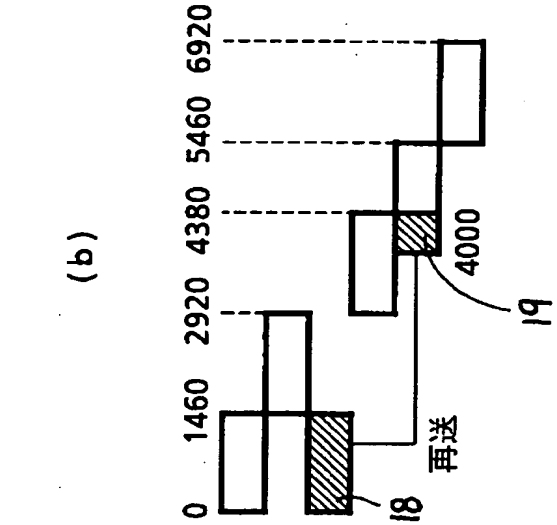
【図４】

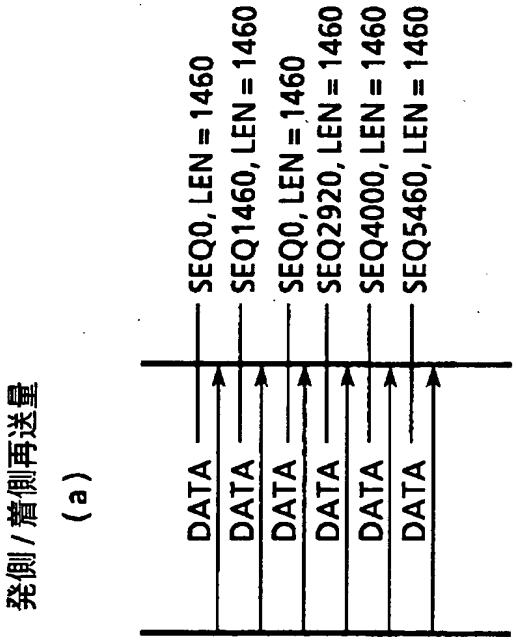


【図５】



【図６】

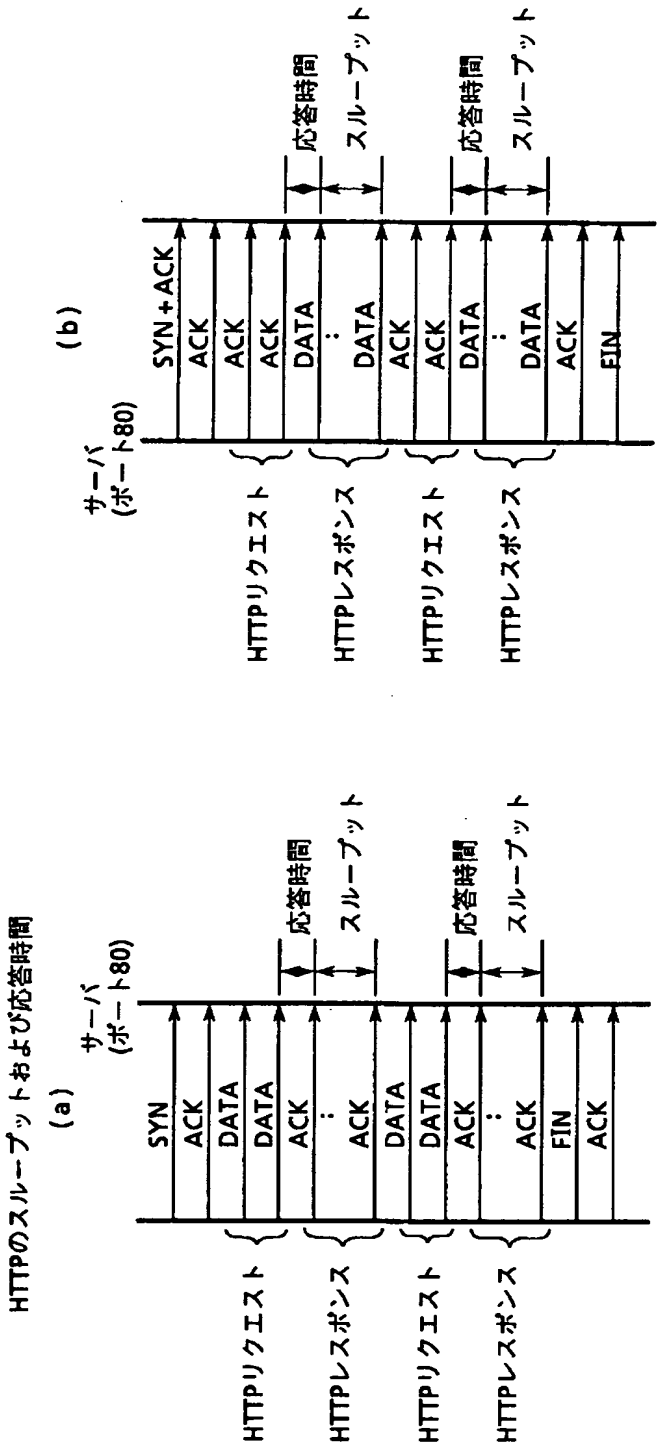

(b)
0
1460
2920
4380
5460
6920
4000
19
18
再送


発側/着側再送量
(a)
DATA
SEQ0, LEN = 1460
SEQ1460, LEN = 1460
SEQ0, LEN = 1460
SEQ2920, LEN = 1460
SEQ4000, LEN = 1460
SEQ5460, LEN = 1460

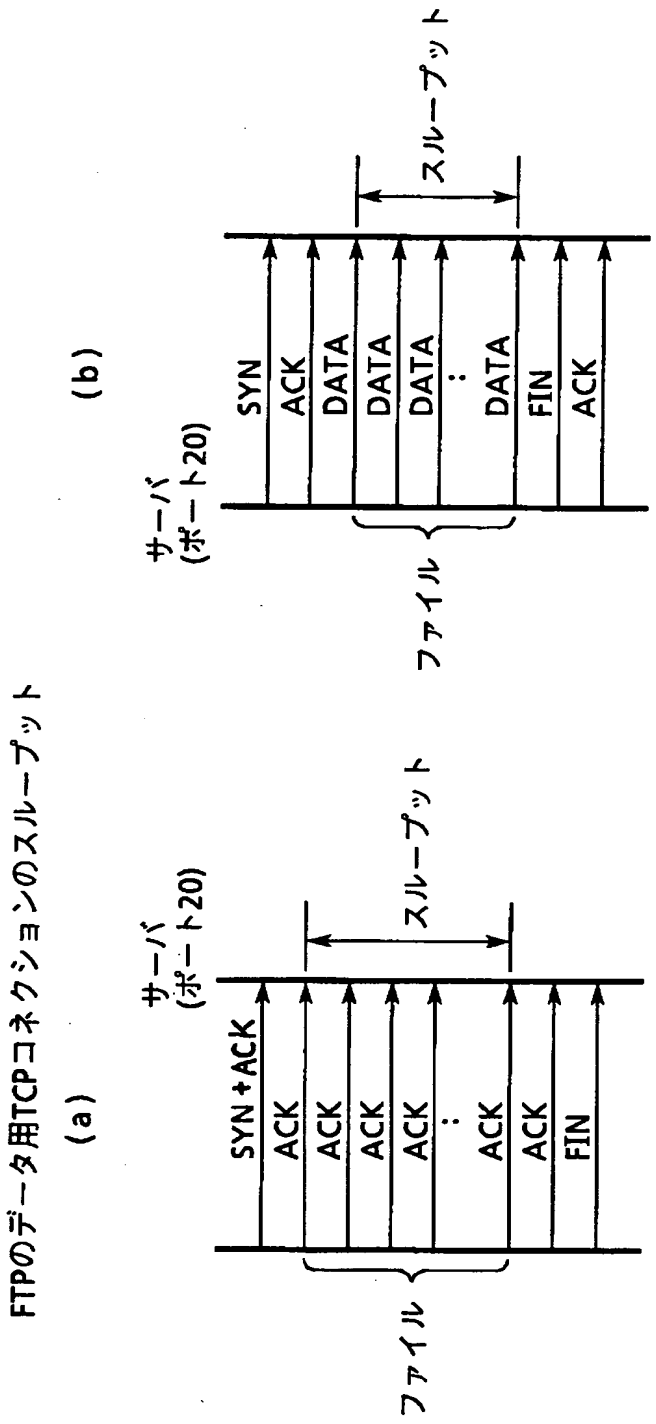
【図７】


発側/着側DTセグメント紛失数
ACK
ACK
ACK
ACK
ACK
ACK
ACK
ACK
ACK=1460, WIN=8192
ACK=2920, WIN=8192
ACK=2920, WIN=8192
ACK = 2920, WIN = 8192
ACK = 4380, WIN = 8192
ACK=4380, WIN=8192
ACK=5840, WIN=8192
ACK = 5840, WIN = 16384
DATAセグメント紛失
DATAセグメント紛失

【図８】


HTTPのスループットおよび応答時間
(a)
サーバ
(ポート80)
SYN
ACK
DATA
DATA
ACK
:
ACK
DATA
DATA
ACK
:
ACK
FIN
ACK
応答時間
スループット
応答時間
スループット
HTTPリクエスト
HTTPレスポンス
HTTPリクエスト
HTTPレスポンス
(b)
サーバ
(ポート80)
SYN+ACK
ACK
ACK
ACK
DATA
:
DATA
ACK
ACK
DATA
:
DATA
ACK
FIN
応答時間
スループット
応答時間
スループット
HTTPリクエスト
HTTPレスポンス
HTTPリクエスト
HTTPレスポンス

【図９】


FTPのデータ用TCPコネクションのスループット
(a)
サーバ
(ポート20)
SYN+ACK
ACK
ACK
ACK
ACK
:
ACK
ACK
FIN
スループット
ファイル
(b)
サーバ
(ポート20)
SYN
ACK
DATA
DATA
DATA
:
DATA
FIN
ACK
スループット
ファイル

【書類名】　　要約書

【要約】

【課題】　片方向のトラヒックしか取得できない場合においても、トラヒックが取得できなかった方向のトラヒック統計情報を収集できること。

【解決手段】　ＳＹＮセグメントが検出された場合、連続して検出されたＤＡＴＡセグメント数を計算し、発側転送セグメント数とする。また、連続して検出されたＤＡＴＡセグメントのうち、最初のＤＡＴＡセグメントのシーケンス番号ＳＥＱ１と、最後のＤＡＴＡセグメントのシーケンス番号ＳＥＱｎと、最後のＤＡＴＡセグメントのユーザデータ長ＬＥＮｎを用い、ｉｎｉ＿ｓｄｔ＝（ＳＥＱｎ＋ＬＥＮｎ）－ＳＥＱ１を計算し、発側転送バイト数ｉｎｉ＿ｓｄｔを求める。

【選択図】　　図２

出　願　人　履　歴　情　報

識別番号　　〔０００００１２１４〕

１．変更年月日　　１９９８年１２月　３日
［変更理由］　名称変更
住　所　　東京都新宿区西新宿２丁目３番２号
氏　名　　ケイディディ株式会社